

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **10283241 A**

(43) Date of publication of application: **23 . 10 . 98**

(51) Int. Cl. **G06F 12/00**
G06F 11/30

(21) Application number: **09088991**

(71) Applicant: **MEIDENSHA CORP**

(22) Date of filing: **08 . 04 . 97**

(72) Inventor: **TSUKUI TETSUYA**

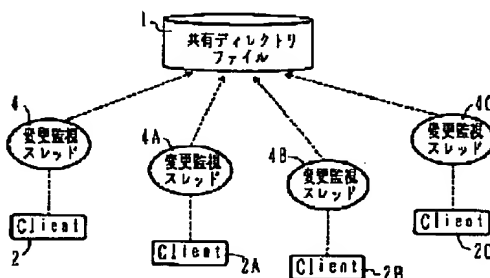
(54) MAINTENANCE SYSTEM FOR MONITOR SYSTEM

(57) Abstract:

PROBLEM TO BE SOLVED: To provide a maintenance system which efficiently performs the maintenance work in parallel with the monitor work with little of extension of equipment.

SOLUTION: A shared file 1 is provided which has information, which is shared among clients 2 and 2A to 2C and is the data maintenance object, in a shared directory and clients are provided with intra-directory file change monitor thread functions 4 and 4A to 4C which are run in parallel with client processes, and each client senses the data change of the shared file due to another client by the change monitor thread and receives a file change report.

COPYRIGHT: (C)1998,JPO



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-283241

(43) 公開日 平成10年(1998)10月23日

(51) Int.Cl.⁶

G 0 6 F 12/00
11/30

識別記号

5 3 3

F I

G 0 6 F 12/00
11/30

5 3 3 J
E

審査請求 未請求 請求項の数1 O L (全 4 頁)

(21) 出願番号 特願平9-88991

(22) 出願日 平成9年(1997)4月8日

(71) 出願人 000006105

株式会社明電舎

東京都品川区大崎2丁目1番17号

(72) 発明者 津久井 哲也

東京都品川区大崎2丁目1番17号 株式会
社明電舎内

(74) 代理人 弁理士 志賀 富士弥 (外1名)

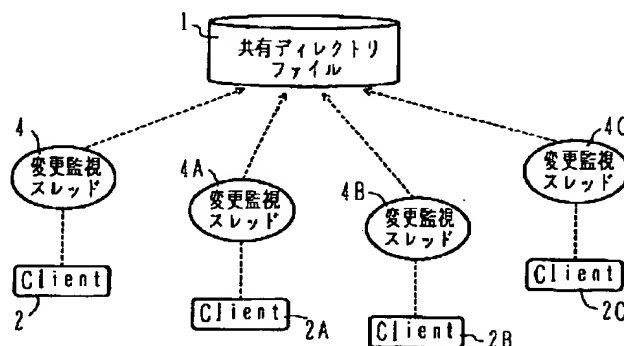
(54) 【発明の名称】 監視システムのメンテナンス方式

(57) 【要約】

【課題】 システムの監視業務とメンテナンス業務を並行して行い、クライアント/サーバ方式で共有データのメンテナンスを行うのでは、サーバプロセス等を必要とする。

【解決手段】 各クライアント2、2A~2Cに共有でデータメンテナンス対象となる情報を共有ディレクトリに置く共有ファイル1を設け、各クライアントは、クライアントプロセスと並行して走るディレクトリ内ファイル変更監視スレッド機能4、4A~4Cを設け、他のクライアントによる共有ファイルのデータ変更を該変更監視スレッドにより感知してファイル変更通知を受ける。

変更監視スレッドによるファイル変更監視システム



【特許請求の範囲】

【請求項1】 ネットワーク上の複数ユーザの作業環境に対応するための機能を持つオペレーティングシステムを搭載した各クライアントにより監視業務とデータメンテナンス業務を行う監視システムにおいて、前記各クライアントに共有でデータメンテナンス対象となる情報を共有ディレクトリに置く共有ファイルを設け、前記各クライアントは、クライアントプロセスと並行して走るディレクトリ内ファイル変更監視スレッド機能を設け、他のクライアントによる前記共有ファイルのデータ変更を該変更監視スレッドにより感知してファイル変更通知を受ける構成にしたことを特徴とする監視システムのメンテナンス方式。

【発明の詳細な説明】**【0001】**

【発明の属する技術分野】 本発明は、監視システム又は監視制御システムにおけるメンテナンス方式に係り、特にコンピュータのグループウェア実現によるデータメンテナンス方式に関する。

【0002】

【従来の技術】 監視システム又は監視制御システム（以下、監視システムで説明する）で行われる業務の1つにデータメンテナンスがある。これはシステムの監視対象である設備などの構成・属性情報などを、実際の運用形態の変更にあわせて更新していく作業である。

【0003】 監視システムの種類によっては、この業務がかなりのウェイトを占めるものもある。例えば電力配電系統監視システムでは、配電設備のメンテナンスの頻度が高いため、データメンテナンス作業の効率が運用コストにかなり響いてくるものとなる。

【0004】 現存の配電系統監視システムでは、操作卓は監視業務用とデータメンテナンス作業用で共通であり、一般的なもので2卓程度にされる。

【0005】

【発明が解決しようとする課題】 監視業務とメンテナンス業務の優先度を考慮すると、メンテナンス業務のみに操作卓を占有させることはできない。したがって、どんなに多くのメンテナンス作業の必要が生じても、1つの操作卓で一人の作業員だけが作業を行わなければならない。

【0006】 このため、一人の作業員が監視業務のかたわらメンテナンス業務を行うことになり、メンテナンス業務の遅れや効率の低下を招く。これには、メンテナンス業務専用の操作卓を設けることが考えられるが、設備コストを高くしてしまう。また、クライアント/サーバ方式で共有データのメンテナンスを行うことが考えられるが、サーバを必要とする。

【0007】 また、大規模なメンテナンス作業は、システムのメーカーに差し戻して行われているため、このため

の運用コストが無視できないものとなっている。

【0008】 本発明の目的は、設備の増設をほとんど不要にしながら監視業務に並行してメンテナンス業務を効率良く行うことができるメンテナンス方式を提供することにある。

【0009】

【課題を解決するための手段】 本発明は、共有ファイルを有してネットワーク構成する監視システムにおいて、各クライアントにはクライアントプロセスと並行して走るディレクトリ内ファイル変更監視スレッドを設けることにより共有ファイル変更通知機能を実現し、これによりサーバプロセスなしで分散データメンテナンス環境を実現するもので、以下の構成を特徴とする。

【0010】 ネットワーク上の複数ユーザの作業環境に対応するための機能を持つオペレーティングシステムを搭載した各クライアントにより監視業務とデータメンテナンス業務を行う監視システムにおいて、前記各クライアントに共有でデータメンテナンス対象となる情報を共有ディレクトリに置く共有ファイルを設け、前記各クライアントは、クライアントプロセスと並行して走るディレクトリ内ファイル変更監視スレッド機能を設け、他のクライアントによる前記共有ファイルのデータ変更を該変更監視スレッドにより感知してファイル変更通知を受ける構成にしたことを特徴とする。

【0011】

【発明の実施の形態】 本実施形態は、監視システムとして、複数の作業員が同時に作業できる「分散データメンテナンス環境」とする。

【0012】 分散環境では一般に、その共有情報を管理するためのアプリケーション（サーバプロセス）が必要であり、全体としてクライアント/サーバ的な構成が要求される。

【0013】 図4は、クライアント/サーバ構成を示し、共有ファイル1に運用設備データ（共有情報）を保存し、クライアント2からサーバ3に属性変更要求をし、サーバ3が共有ファイル1の当該データの属性変更を行う場合、クライアント2はサーバプロセスに対し、「属性変更」通知をプロセス間通信により行い、それによりサーバ3は共有情報の属性変更をおこなった後、接続している全てのクライアント2A、2B、2Cに、共有情報に変更されたという通知を行うこととなる。その他、複雑な管理機能を要求される場合にはサーバは必須のものとなる。

【0014】 本実施形態は、サーバプロセスに要求される機能を本質的なものに制限することによって、コンピュータに搭載されるOS（オペレーティングシステム）の機能を利用したサーバプロセスなしの分散環境を実現する。

【0015】 以下、監視システムのLANにMicrosoft社のWindows NTを載せ、PC接続した

環境を本実施形態として説明する。

【0016】Windows NTでは、ネットワーク上の複数ユーザの作業環境に対応するための機能（ネットワーク共有ファイルシステム、ユーザアカウントなど）がOSに既に実装されているため、分散環境で必要とされる機能のかなりの部分の開発を省くことができる。

【0017】そこで、Windows NTのOSの機能をうまく使うことでサーバプロセスなしに分散環境を実現する。本実施形態では特にこれを「共有情報変更通知」のみに限定する。

【0018】本実施形態の構成を図1に示す。各クライアント2、2A～2Cが共有する情報は、共有ファイル1の特定のディレクトリに置く。Windows NTには、ディレクトリ内のファイルの属性（名前、ファイル属性、サイズ、タイムスタンプ、セキュリティ情報）の変更を監視する機能がある。これは、Windows NTの「変更通知オブジェクト」を生成してそれが発するシグナルを待つ、という形で実現する。プロシージャ例を図2に示す。

【0019】各クライアント2、2A～2Cは、このプロセスを、自プロセスの1つの変更監視スレッド4、4A～4Cとして常に並行に走らせておく。このプロセス中の「変更通知処理」の部分にクライアント自体へのメッセージ送信コードを記述して、クライアントプロセスへの共有ファイル変更通知機能を実現する。

【0020】図3は、図1の構成における変更監視スレッドによる共有情報の変更手順を示し、クライアント2が共有ファイル1のデータを変更すると、起動している他のすべてのクライアント2A～2Cの変更監視スレッド4A～4Cがそれを「感知」し、各クライアントは共有ファイルの変更を知ることができる。

【0021】これにより、図4のクライアント／サーバ方式とまったく同じ機能がクライアントのみで実現でき*

*る。「変更監視スレッド」はちょうど「共有情報変更通知」機能のみをもつサーバとしての役割を持つ。

【0022】

【発明の効果】以上のとおり、本発明によれば、各クライアントにはクライアントプロセスと並行して走るディレクトリ内ファイル変更監視スレッドを設けることにより共有ファイル変更通知機能を実現するため、サーバプロセスなしで分散データメンテナンス環境を実現することができ、以下の効果がある。

10 【0023】（1）複数の作業者が同時に作業できる「分散データメンテナンス環境」により、メンテナンス業務の効率化が期待できる。

【0024】（2）Windows NTなどコンピュータのOSの機能をうまく利用すれば、サーバを作ることなしに簡易に分散データメンテナンス環境を実現でき、サーバ作成分の工数をまったくカットすることができる。

【0025】（3）プロセス間通信の機能を使わないで実現できるため、その分のクライアントアプリケーションの実装工数もかからない。

【図面の簡単な説明】

【図1】本発明の実施形態を示すファイル変更監視システム。

【図2】実施形態における変更監視プロセスの例。

【図3】実施形態における変更監視スレッドによる共有情報の変更手順。

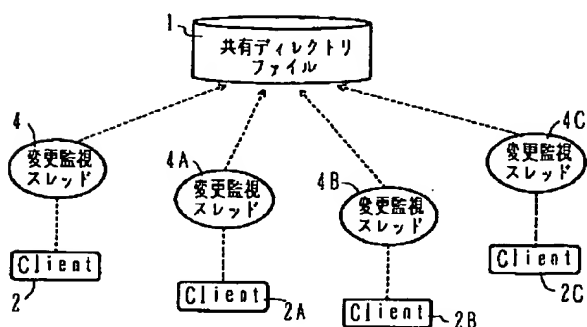
【図4】クライアント／サーバ方式の共有情報の変更手順。

【符号の説明】

- 1…共有ファイル
2、2A～2C…クライアント
3…サーバ
4、4A～4C…変更監視スレッド

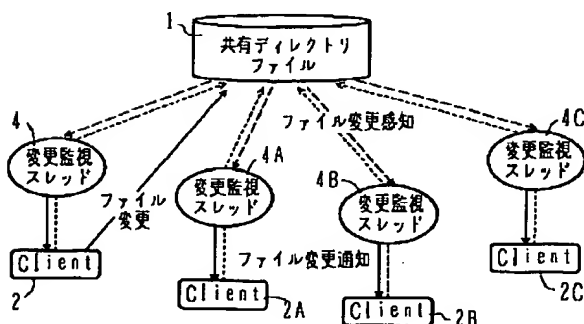
【図1】

変更監視スレッドによるファイル変更監視システム



【図3】

変更監視スレッドによる共有情報の変更手段



【図2】

変更監視プロセス+例

```

UNIT WatchProc(LPVOID pParam)
{
    HANDLE hChange = ::FindFirstChangeNotification(
        "共有ファイルが置かれているディレクトリへのパス",
        FALSE,
        FILE_NOTIFY_CHANGE_LAST_WRITE    //ファイルのタイムスタンプが変わったとき
    ); //シグナルを発する「オブジェクト」を作る

    while(TRUE) {
        //変更通知待機
        //シグナルを発するまで制御が戻らない
        DWORD status = ::WaitForSingleObject(hChange, INFINITE);

        if (status == WAIT_OBJECT_0) {
            /*変更通知処理をここに書く
            .....*/
        }

        //変更通知待機継続
        if (::FindNextChangeNotification(hChange) == FALSE) {
            return 0;
        }
    }
}

```

【図4】

共有情報の変更手順

